TD7

Exercice 1

Soit la base STATION DE SKI de schéma:

hotel (noms,nomh,categorie,adresse,tel,nb_chambres)
station (noms,gare)
activite (type_activite,noms)

Pour chacune des requêtes suivantes, on demande:

1. l'arbre syntaxique de la requête
2. le plan d'exécution obtenu par la restructuration algébrique.

Les principes qu'il faut prendre en compte pour la restructuration algébrique sont les suivants:

a. évaluer les sélections (ou restrictions) le plus tôt possible. En effet, la relation qu'on obtient par l'évaluation de sélections est plus petite que la relation initiale.
b. faire des projections pour réduire la taille de la relation en question.
c. permuter les jointures quand c'est nécessaire.

Requête 1 :

adresse, numéro de téléphone et nombre de chambres des hôtels de catégorie 3' dans la station de nom (noms = 'persey'.)

```
SELECT adresse, tel, nb_chambres
FROM hotel
WHERE noms='pesey' AND categorie=3;
```

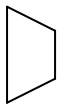
Requête 2 :

nom de station (noms) et la gare de la station pour les stations ayant pour activité le tennis.

```
SELECT noms, gare
FROM station, activite
WHERE type_activite = ``tennis''
AND station.noms=activite.noms
```

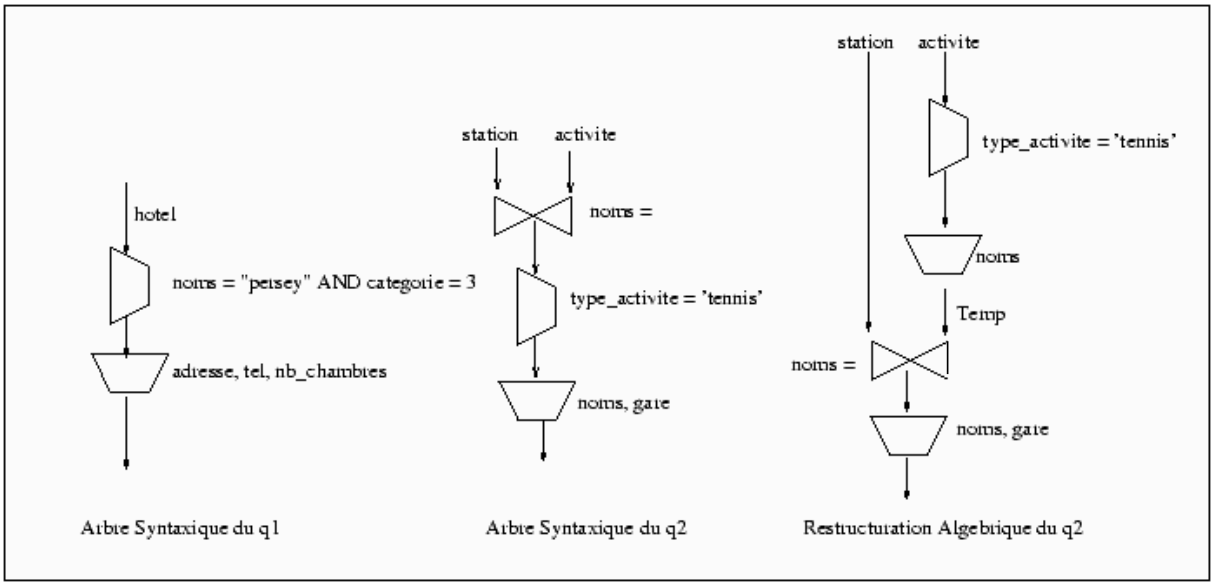
Utilisez les symboles suivants :

Sélection :



Projection :

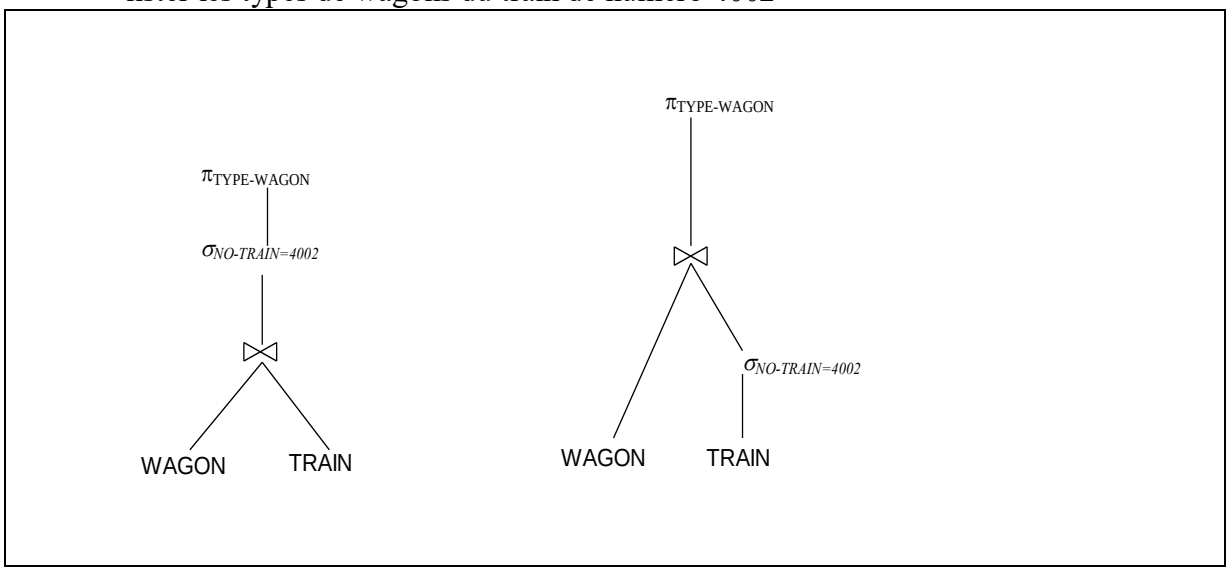
Jointure :



Exercice 2

1. On considère les deux relations suivantes :
   TRAIN(NO-TRAIN, NO-WAGON)
   WAGON(NO-WAGON, TYPE-WAGON, POIDS-VIDE, CAPACITE, ETAT,GARE)

Donner deux arbres graphiques différents pour la requête :
''lister les types de wagons du train de numéro 4002''



Exercice 3

Sur la base de données suivante :
PERSONNE (NumPer, Nom, Prénom, Age)

Décrivant des personnes en donnant leur numéro, nom, prénom et age

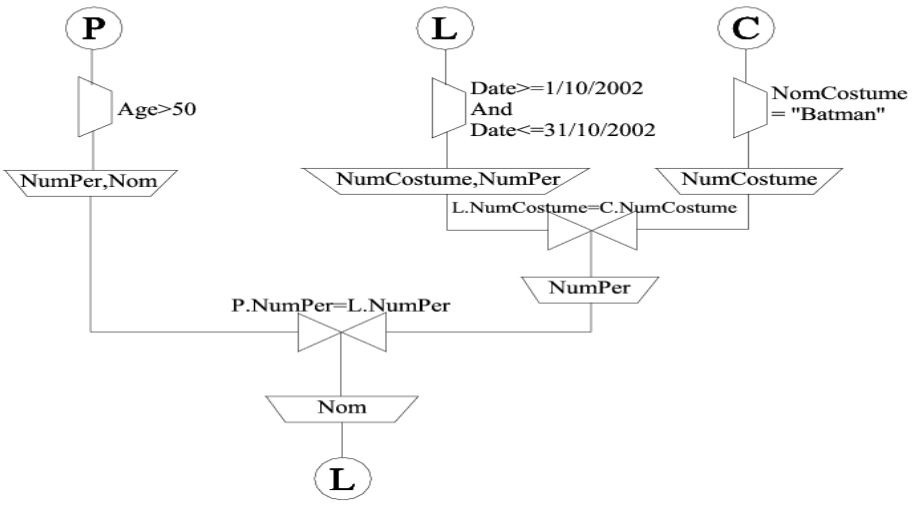
COSTUME (NumCostume, NomCostume, Taille)

Décrivant des costumes à louer, en donnant leur numéro, leur nom, et la taille qui peut prendre les valeurs : ’4-6’, ’8-10’, ’12-14’, ’S’, ’M’, ’L’

LOCATION (NumCostume, NumPer, DateLocation, DuréeLocation)

DateLocation est donnée sous forme jj/mm/aa

Nous voulons exprimer la requête suivante :
Rechercher le nom des personnes de plus de 50 ans qui ont loué un costume de ’’Batman’’ (nom du costume) au mois d’octobre 2002.
Proposez un arbre d’opérateurs de l’algèbre relationnelle optimisé pour l’exécution de cette requête. ***(3 points)***



Exercice 4

Considérons une base de données contenant des informations utiles pour une compagnie de téléphone. La base de données maintient les tables suivantes :

- ***Client(N°_téléphone, Nom_ client)***, qui contient des informations sur les clients.
- ***Plans_Appels(Plan_Id, Plan_nom)***, qui contient des informations sur les différents plans d'appels offérent par la compagnie.
- ***Appels(De, Vers, Temps, Jour, Mois, Année, Durée, Plan_Id, Coût)***, qui contient des informations sur chaque appel.

Supposons que la compagnie de téléphone s'intéresse à déterminer les plans d'appels qui ont rapporté plus de 1 millions d'Euros au cours de l'une des années comprises entre 1990 et 1995. La requête SQL suivante peut être utilisée pour exprimer ce besoin :

```
Q :   SELECT Année, Plan_nom, SUM(Coût)
      FROM Appels, Plans_Appels
      WHERE Appels.Plan_Id = Plans_Appels.Plan_Id
            AND Année >= 1990 AND Année <= 1995
      GROUPBY Année, Plan_nom
      HAVING SUM(Coût)> 1,000,000
```

La compagnie téléphonique maintient également des vues matérialisées qui résument la performance de chacun de ses plans d'appels sur des périodes choisies. En particulier, nous supposons que la vue matérialisée suivante V1(Plan_Id, Mois, Année, Gains) est disponible :

```
V1 :  SELECT Plan_Id, Mois, Année, SUM(Coût)
      FROM Appels
      GROUPBY Plan_Id, Mois, Année
```

Donnez une réécriture de Q en utilisant V1.

```
Q' :  SELECT Année, Plan_nom, SUM(Gains)
      FROM V1, Plans_Appels
      WHERE V1.Plan_Id = Plans_Appels.Plan_Id
            AND Année >= 1990 AND Année <= 1995
      GROUPBY Année, Plan_nom
      HAVING SUM(Gains)> 1,000,000
```